Claims

What is claimed is:

1. A non-disruptive method for replacing a first software module of a system with a second software module, the method comprising:

copying update control code from a first software module of a system to memory space outside a memory location of the first software module;

replacing the first software module with a second software module by storing the second software module in memory at a location which at least partially overlies the first software module, wherein the replacing includes employing the update control code copied from the first software module to facilitating the replacing; and

beginning execution of the second software module without resetting the system.

2. The method of claim 1, wherein the replacing includes executing the update control code copied from the first software module during the replacing of the first software module with the second software module.

3. The method of claim 1, wherein the first software module comprises a firmware module, and the system comprises an embedded system, and wherein the replacing includes overlaying the memory location of the firmware module with the second software module, the second software module comprising an updated firmware module.

4. The method of claim 1, wherein the update control code includes update control code for monitoring replacing of the first software module with the second software module.

5. The method of claim 4, wherein the update control code further includes control code for branching to an entry point of the second software module upon completion of the replacing to facilitate the beginning execution of the second software module.

6. The method of claim 1, wherein the first software module includes at least one of a loader and a linker, and wherein the replacing includes overlaying the memory location of the first software module with the second software module.

7. The method of claim 1, wherein the first software module and the second software module each comprise a single statically linked module.

8. The method of claim 1, further comprising storing data to be maintained during the replacing of the first software module to memory space outside the memory location of the first software module.

9. The method of claim 1, wherein the system comprises an embedded system, and the first software module and the second software module each comprise a firmware module.

10. The method of claim 1, wherein the replacing employs a hardware based direct memory access (DMA) operation to save the second software module to a target memory space and wherein the copying update control code comprises copying the update control code to memory space outside the target memory space, and wherein the update control code includes control code for determining when the DMA operation has completed and for branching to an entry point of the second software module upon completion of the DMA operation.

11. A non-disruptive system for replacing a first software module of a computing system with a second software module, the non-disruptive system comprising:

means for copying update control code from a first software module of a computing system to memory space outside a memory location of the first software module;

means for replacing the first software module with a second software module by storing the second software module in memory at a location which at least partially overlies the first software module, wherein the means for replacing includes means for employing the update control code copied from the first software module to facilitate the replacing; and

means for beginning execution of the second software module without resetting the computing system, thereby providing non-disruptive replacing of the first software module with the second software module.

12. The system of claim 11, wherein the means for replacing includes means for executing the update control code copied from the first software module during the replacing of the first software module with the second software module.

13. The system of claim 11, wherein the first software module comprises a firmware module, and the computing system comprises an embedded system, and wherein the means for replacing includes means for overlaying the memory location of the firmware module with the second software module, the second software module comprising an updated firmware module.

14. The system of claim 11, wherein the update control code includes update control code for monitoring replacing of the first software module with the second software module.

15. The system of claim 14, wherein the update control code further includes control code for branching to an entry point of the second software module upon completion of the replacing to facilitate the beginning execution of the second software module.

16. The system of claim 11, wherein the first software module includes at least one of a loader and a linker, and wherein the means for replacing includes means for overlaying the memory location of the first software module with the second software module.

17. The system of claim 11, wherein the first software module and the second software module each comprise a single statically linked module.

18. The system of claim 11, further comprising means for storing data to be maintained during the replacing of the first software module to memory space outside the memory location of the first software module.

19. The system of claim 11, wherein the computing system comprises an embedded system, and the first software module and the second software module each comprise a firmware module.

20. The system of claim 11, wherein the means for replacing comprises a hardware based direct memory access (DMA) operation to save the second software module to a target memory space and wherein the means for copying update control code comprises means for copying the update control code to memory space outside the target memory space, and wherein the update control code includes control code for determining when the DMA operation has completed and for branching to an entry point of the second software module upon completion of the DMA operation.

21. A non-disruptive system for replacing a first software module of a computing system with a second software module, the non-disruptive system comprising:

a first memory space containing a first software module of a computing system, and a memory location outside the first memory space for receiving update control code copied from a portion of the first software module;

a controller for replacing the first software module with a second software module by storing the second software module in a second memory space at a location which at least partially overlies the first memory space containing the first software module, wherein the replacing includes employing the update control code copied from the first software module to facilitate the replacing; and

wherein the controller is adapted to begin execution of the second software module without resetting the computing system.

22. At least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a non-disruptive method for replacing a first software module of a system with a second software module, the method comprising:

copying update control code from a first software module of a system to memory space outside a memory location of the first software module;

replacing the first software module with a second software module by storing the second software module in memory at a location which at least partially overlies the first software module, wherein the replacing includes employing the update control code copied from the first software module to facilitating the replacing; and

beginning execution of the second software module without resetting the system.

23. The at least one program storage device of claim 22, wherein the replacing includes executing the update control code copied from the first software module during the replacing of the first software module with the second software module.

24. The at least one program storage device of claim 22, wherein the first software module comprises a firmware module, and the system comprises an embedded system, and wherein the replacing includes overlaying the memory location of the firmware module with the second software module, the second software module comprising an updated firmware module.

25. The at least one program storage device of claim 22, wherein the update control code includes update control code for monitoring replacing of the first software module with the second software module.

26. The at least one program storage device of claim 25, wherein the update control code further includes control code for branching to an entry point of the second software module upon completion of the replacing to facilitate the beginning execution of the second software module.

27. The at least one program storage device of claim 22, wherein the first software module includes at least one of a loader and a linker, and wherein the replacing includes overlaying the memory location of the first software module with the second software module.

28. The at least one program storage device of claim 22, wherein the first software module and the second software module each comprise a single statically linked module.

29. The at least one program storage device of claim 22, wherein said method further comprises storing data to be maintained during the replacing of the first software module to memory space outside the memory location of the first software module.

30. The at least one program storage device of claim 22, wherein the system comprises an embedded system, and the first software module and the second software module each comprise a firmware module.

31. The at least one program storage device of claim 22, wherein the replacing employs a hardware based direct memory access (DMA) operation to save the second software module to a target memory space and wherein the copying update control code comprises copying the update control code to memory space outside the target memory space, and wherein the update control code includes control code for determining when the DMA operation has completed and for branching to an entry point of the second software module upon completion of the DMA operation.

* * * * *